Linear Projections and Gaussian Process Reconstructions

Joaquin Quiñonero-Candela[1]
Neil D. Lawrence[2] Carl E. Rasmussen[3]

[1]Technical University of Berlin and Fraunhofer FIRST.IDA
(Sept-Dec 2007 visiting Universidad Carlos III de Madrid)
(from January 2007 Microsoft Research Cambridge)

[2]University of Sheffield
(from January 2007 University of Manchester)

[3]Max Planck Institute for Biological Cybernetics
(from April 2007 Cambridge University)

Learning06 - Vilanova i la Geltrú
Tuesday October 3rd, 2006

Acknowledgements

Thanks PASCAL for funding visit of JQC to NL in Sheffield, in the summer 2005, where the back-constraints idea was cooked up

Linear Dimensionality Reduction

Dimensionality reduction: $D \gg q$

- Consider high-dimensional data $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_N]$ in $\mathcal{R}^D$
- low dimensional latent representation $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$ in $\mathcal{R}^q$

Linear Projection

- Find a matrix $\mathbf{P}$ of size $q \times D$ and project

$$\mathbf{x}_i = \mathbf{P}\,\mathbf{y}_i$$

- Standard choice are principal components of data (PCA)
- Rows of $\mathbf{P}$ are the first q eigenvectors of $\mathbf{Y}\mathbf{Y}^\top$ (up to scaling)
- Minimum mean squared reconstruction error

Linear Reconstructions

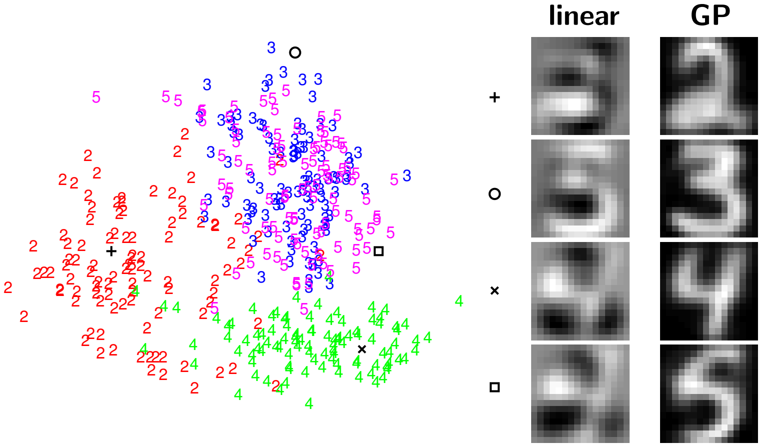
Linear map from latent to data

- The reconstruction of the $\mathbf{y}_i$ from the $\mathbf{x}_i$ is also linear
- Reconstructed hyperplane is spanned by principal eigenvectors
- This is often a poor reconstruction!
- But most dimensional reduction methods don't even offer a map between latent and data

Example: hand-written digits

- 16×16 gray-scale images of the 2, 3, 4 and 5s
- 2-dimensional PCA projection
- Linear reconstruction from PCA

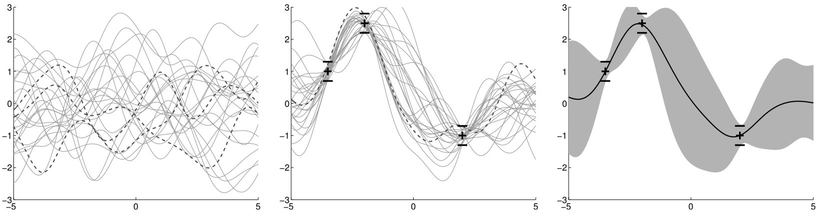
A Poor Reconstruction vs a Cool Reconstruction



Reconstruction as a Regression Problem

- Once we have linearly projected, we have a set of pairs of inputs and outputs $\{\mathbf{x}_i, \mathbf{y}_i\}$
- Learn a mapping through non-linear regression!

Bayesian Regression with Gaussian Process Priors



left samples from our prior, a Gaussian Process

middle samples from the posterior, data observed (crosses) and uniform noise model (horizontal bars)

right predictive distribution, empirically computed from the posterior samples. Here mean and 2 std dev given

parameters of the prior? Either specify hyperprior on, or learn the parameters of the prior by maximizing the **evidence**

Gaussian Processes as Smooth Priors Over Functions

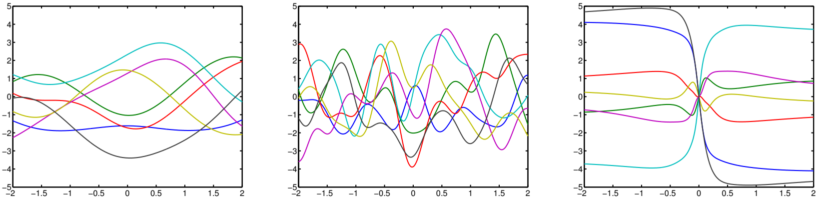
Smoothness enforcing priors

- if $\mathbf{x}_i$ and $\mathbf{x}_j$ are similar, then $f(\mathbf{x}_i)$ and $f(\mathbf{x}_j)$ are similar

$$p\left(\begin{bmatrix} f(\mathbf{x}_i) \\ f(\mathbf{x}_j) \end{bmatrix} \middle| \mathbf{x}_i, \mathbf{x}_j, \theta\right) = \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{K}_{ii} & \mathbf{K}_{ij} \\ \mathbf{K}_{ij} & \mathbf{K}_{jj} \end{bmatrix}\right)$$

- Covariance function determines kind of smoothness, example:

$$\mathbf{K}_{ij} = \mathrm{Cov}\left\{f(\mathbf{x}_i), f(\mathbf{x}_j)\right\} = k(\mathbf{x}_i, \mathbf{x}_j, \theta) = v^2 \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\lambda^2}\right)$$



Evidence and predictive distribution

- Assuming an independent Gaussian noise model

$$y_i = f(\mathbf{x}_i) + \epsilon_i \qquad \epsilon_i \sim \mathcal{N}(0, \sigma^2) \qquad p(\mathbf{y}|\mathbf{f}) = \mathcal{N}(\mathbf{f}, \sigma^2\,\mathbf{I})$$

- the evidence is a Gaussian Process as well

$$p(\mathbf{y}|\mathbf{X}, \theta) = \int p(\mathbf{y}|\mathbf{f})\, p(\mathbf{f}|\mathbf{X}, \theta)\mathrm{d}\mathbf{f} = \mathcal{N}(\mathbf{0}, \mathbf{K} + \sigma^2\,\mathbf{I})$$

- the predictive distribution at a new input $\mathbf{x}_*$ is a Gaussian too

$$p(f(\mathbf{x}_*)|\mathbf{x}_*, \mathbf{X}, \mathbf{y}, \theta) = \mathcal{N}(m_*, v_*)$$

$$m_* = K_{*,N}\,[\mathbf{K}_{N,N} + \sigma^2\,\mathbf{I}]^{-1}\mathbf{y}$$
$$v_* = K_{*,*} - K_{*,N}\,[\mathbf{K}_{N,N} + \sigma^2\,\mathbf{I}]^{-1}\,K_{N,*}$$

Gaussian Process Latent Variable Model (GP-LVM)

- Until now I have been given the embedding $\mathbf{X}$
- In addition to reconstructing, can I also learn the embedding?

A product of GPs model (Lawrence, NIPS 16, 2004)

- Predict each dimension of $\mathbf{Y}$ with an independent GP
- Take $\mathbf{X}$ to be the common inputs to all D regression models

$$p(\mathbf{Y}|\mathbf{X},\theta) = \prod_{d=1}^{D} p(\mathbf{y}^d|\mathbf{X},\theta)$$

- learn the inputs $\mathbf{X}$ (and the hyperparameters θ)

Gaussian Process Latent Variable Model (GP-LVM)

- Until now I have been given the embedding $\mathbf{X}$
- In addition to reconstructing, can I also learn the embedding?

A product of GPs model (Lawrence, NIPS 16, 2004)

- Predict each dimension of $\mathbf{Y}$ with an independent GP
- Take $\mathbf{X}$ to be the common inputs to all D regression models

$$p(\mathbf{Y}|\mathbf{X}, \theta) = \prod_{d=1}^{D} p(\mathbf{y}^d|\mathbf{X}, \theta)$$

- learn the inputs $\mathbf{X}$ (and the hyperparameters θ)

The GP-LVM in action

Motion capture data

- Subject breaking into a run from standing
- Data dimension: 102, 3D position of 34 markers
- Data from Ohio State University Advanced Computing Center for the Arts and Design

`http://accad.osu.edu/research/mocap/mocap_data.htm`

Strength of the GP-LVM

- A powerful, probabilistic reconstruction mapping from latent to data space

Limitations of the GP-LVM

- Optimization in a large space (dim at least $N \times q$)
- There are extremely many local optima (initialize carefully)
- No explicit mapping from data to latent space
- The GP-LVM is is not similarity preserving

The GP-LVM is **dissimilarity preserving** (a limitation?)

- Because it is a smooth mapping from $\mathbf{X}$ to $\mathbf{Y}$
- Advantage of avoiding overlapping effect (LLE, Isomap, etc)
- Less sensitive to noise than local similarity preserving embeddings
- Inability to preserve local structure in the data
  $\rightarrow$ Lawrence initializes with PCA!

Symbiosis

Linear projections need GP reconstructions, and the GP-LVM needs linear projections

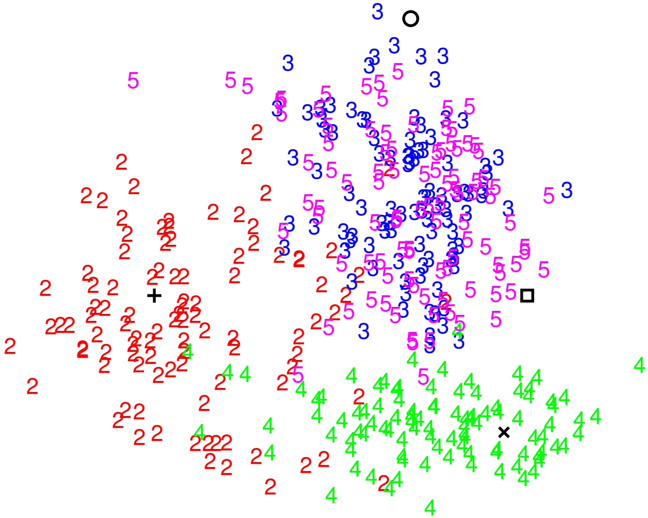
Learn an optimal projection for a GP reconstruction

- Instead of initializing with PCA, why not directly learn the optimal linear projection for GP reconstruction?
- Replace $\mathbf{X}$ by $\mathbf{X} = \mathbf{P}\,\mathbf{Y}$ and learn $\mathbf{P}$ by max GP evidence
- Smaller $q \times D$ optimization space (can init at random)

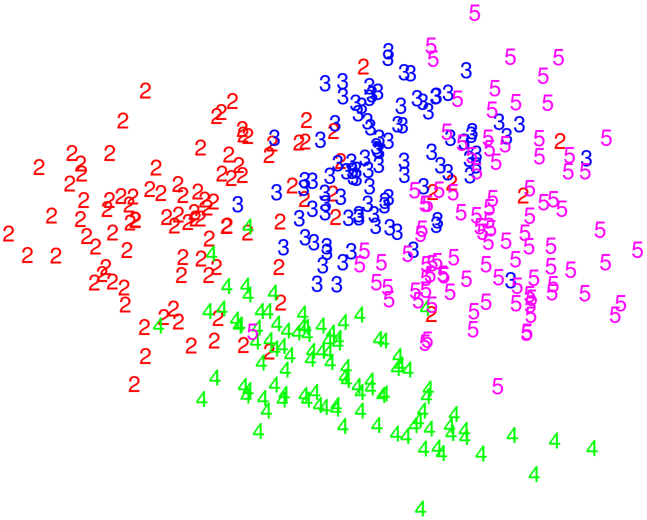
What kind of linear projections do we get?

- More dissimilarity preserving than PCA!
- Examples: motion capture, digits, and swiss roll

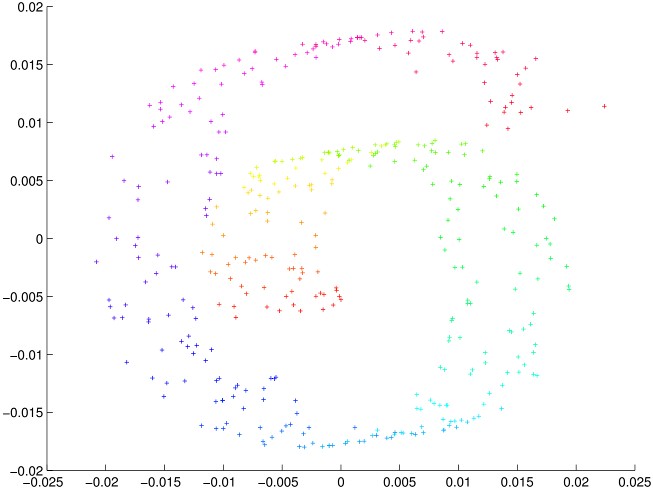
Digits Revisited



Digits Revisited



Swiss Roll


0.02
0.015
0.01
0.005
0
−0.005
−0.01
−0.015
−0.02
−0.025
−0.02
−0.015
−0.01
−0.005
0
0.005
0.01
0.015
0.02
0.025

Discussion

- Powerful, probabilistic generative GP model latent to data
  - Computer animated graphics, imitation learning
  - Prior over poses (tracking, pose recovery) (Growchow et al, SIGGRAPH'03)(Urtasun et al, ICCV'05)
- A linear map from data to latent optimized for GP reconstruction
- Heals the GP-LVM from some of its curses
- Particular case of the back-constrained GP-LVM (Lawrence and Quiñonero-Candela, ICML 2006)
- Is this still a proper probabilistic model?